Fedora 8 + Xen 3.3.0 Installation Guide

1.Installing and configuring Fedora 8

1.1. Installing Fedora 8

1. Boot and start the installation process from DVD-ROM

2. ***(Optional)*** Choose "`Skip`" when you are asked to test the installation media. Testing the media is unnecessary and is time consuming.

3. ***(Recommended)*** In the "Installation Language" step, choose "`English`".

4. In the "Partitioning" step, choose "`Create custom layout`", and click "`Next`" to partition the disk manually.

5. Create at least 3 partitions. The following layout is recommended:

Device	Size	Type	Mount point
/dev/sda1	1 GB	ext3	/boot
/dev/sda2	20 GB	ext3	/
/dev/sda3	4 GB	swap	
/dev/sda4	(MAX)	extended	
/dev/sda5	(MAX)	LVM	

6. ***(Recommended)*** In the "Time zone" step, choose "`Shanghai, Asia`" and UNCHECK the option "`System clock uses UTC`". This step ensures that the time-stamps of files and system logs are in local time.

7. In the "Installation components" step, make sure to select all "`Software Development`" packages. These packages are required for software compiling and kernel hacking.

8. ***(Optional)*** Choose "`Customize now`" to modify the installation components in detail. Here is a list of useful packages which are not installed by default:

```
Applications / Editors / emacs
Applications / Text-based internet / lynx
Development / Development Tools / rpmdevtools
Base System / Base / bridge-utils
Base System / System Tools / {jigdo, ntfs-3g, screen, vnc, yum-utils}
Base System / Virtualization / {kernel-xen, kvm, qemu, xen}
Base System / X Window System / {vnc-server, wqy-bitmap-fonts}
Languages / Chinese Support
```

9. Install the system and then reboot

10. In the welcome screen, make sure to DISABLE `Firewall` and `SELinux`.

Note:
To get the installation image of Fedora 8, checkout the mirror list at http://mirrors.fedoraproject.org/publiclist/Fedora/8/.

1.2. Configuring Fedora 8

1. Login to a shell with the `root` account

2. Update the `yum` configurations

```
# yum update
```

3. ***(Optional)*** Type "`yum update`" again to update the entire system. This step may take plenty of time because it will download approximately 1GB of packages from the Internet. Be patient!

4. ***(Optional)*** During updating, you may modify the following configuration files to make the system easier to use:

   a. Disable graphic GRUB menus and keep the booting process diagnostic

   ```
   # vi /boot/grub/menu.lst
   ```

   Comment out the following two lines:

   ```
          "splashimage=(hd0,0)/grub/splash.xpm.gz"
          "hiddenmenu"
   ```

   Delete the "`rhgb`" and "`quite`" options from each kernel's command line

   b. Change the default run level to 3 to avoid entering the X11 window system every time

   ```
   # vi /etc/inittab
   ```

   Change the line "`id:5:initdefault:`" to "`id:3:initdefault:`"

   In the future, you may start X11 temporarily by executing "`init 5`"

   c. Grant "`sudo`" privileges to users

   ```
   # vi /etc/sudoers
   ```

   Duplicate the line "`root ALL=(ALL) ALL`" and replace "`root`" with the username you would like to grant access to, for example:

   ```
          ## Allow root to run any commands anywhere
          root   ALL=(ALL) ALL
          hchen  ALL=(ALL) ALL
          gis    ALL=(ALL) ALL
   ```

   d. Setup paths and aliases for the root account (otherwise, "`su`" or "`sudo`" do not setup them correctly)

   ```
   # vi /root/.bashrc
   ```

   Append the following lines:

   ```
          PATH=$PATH:/sbin:/usr/sbin
          alias vi='vim'
   ```

   e. Enable bash to search the command history using "`Ctrl-p`" and "`Ctrl-n`"

   ```
   # vi /boot/inputrc
   ```

   Append the following two lines:

   ```
          "\C-p": history-search-backward
          "\C-n": history-search-forward
   ```

   f. Customize the `TAB` behavior for vim

   ```
   # vi /etc/vimrc
   ```

   Append the following lines:

   ```
          set tabsize=4
          set shiftwidth=4
          set expandtab
   ```

   g. Start a VNC server on behalf of a user

   ```
   # su <username>
   $ vncserver :<id>
   ```

   (Type the desktop password twice)

   ```
   $ vncserver -kill :<id>
   $ vi ~/.vnc/xstartup
   ```

Comment out the following lines:

```
vncconfig -iconic &
xterm -geometry 80x24+10+10 -ls -title "$VNCDESKTOP Desktop" &
twm &
```

Then append a line:

```
       exec gnome-session &
$ vncserver :<id>
$ exit
# vi /etc/rc.local
```

Append a line:

```
runuser <username> -c “vncserver :<id>”
```

Now, you can connect to the VNC desktop using any VNC client via “`<server ip>:<id>`”. The server will startup automatically on booting. Note that you may set up several desktops for different users, using distinct `<id>`’s

2.Installing Xen 3.3 and the Domain 0

2.1. Installing the Xen kernel

1. Prepare the source tree

```
$ wget http://bits.xensource.com/oss-xen/release/3.3.0/xen-3.3.0.tar.gz
$ tar -xvzf xen-3.3.0.tar.gz
```

Alternatively, you may checkout the source from our SVN server:

```
$ svn co svn://192.168.1.91/vs/xen/trunk xen-3.3.0
```

2. Compiling the source

```
$ cd xen-3.3.0
$ make xen tools
```

Hint: you can specify the number of concurrent jobs to accelerate the making process. For example, if the platform has 4 cores, you can type “`make xen tools -j 4`” to make full use of them.

3. Installing Xen and user-level tools

```
$ sudo make install-xen install-tools
```

4. Check that the following files exist before moving to the next step:

```
/boot/xen-3.3.0.gz
/usr/sbin/xend
/usr/sbin/xm
```

2.2. Installing the “Xenified” Linux kernel from pre-compiled binaries

1. Login to a shell as `root`

2. Get kernel binaries by `yum`

If you didn’t select package “`kernel-xen`” in step 1.1-8, then install it now:

```
# yum install kernel-xen
```

Or, if you didn’t update the entire system as described in step 1.2-3, then update the “`kernel-xen`” package now:

```
# yum update kernel-xen
```

Normally, you will have a kernel image “`vmlinuz-2.6.21.7-5.fc8xen`” installed in the “`/boot`” directory. Check it before moving on.

3. Create a new init-ram-disk (initrd) for Domain 0 by adding the backend drivers

```
# mkinitrd -v -f --with=ext3 --with=pciback --with=netbk --with=blkbk --
with=netloop --with=xenbus_be --with=netloop /boot/initrd-xen0
2.6.21.7-5.fc8xen
```

Check that the file "`/boot/initrd-xen0`" exists before moving on.

4. Create a new boot entry in GRUB

```
# vi /boot/grub/menu.lst
```

Append the following lines:

```
    title XEN 3.3 with Fedora 8 (2.6.21.7-5.fc8xen)
            root (hd0,0)
            kernel /xen-3.3.0.gz
            module /vmlinuz-2.6.21.7-5.fc8xen ro root=/dev/sda2
            module /initrd-xen0
```

Optionally, you can append "`dom0_mem=xxx`" (`xxx` is in megabytes) at the end of "`kernel`" line to restrict the memory usage of the Domain 0.

5. ***(Optional)*** Enable serial console in Xen *(only applicable when the machine has a PCI serial card installed)*

```
# vi /boot/grub/menu.lst
```

Append the following lines:

```
    title XEN 3.3 with Fedora 8 (2.6.21.7-5.fc8xen) serial-line
            root (hd0,0)
            kernel /xen-3.3.0.gz loglvl=all com1=115200,8n1,0x1030,0
console=com1,vga sync_console conswitch=bx
            module /vmlinuz-2.6.21.7-5.fc8xen ro root=/dev/sda2
console=ttyS0,115200n8 console=tty0
            module /initrd-xen0
```

Note that `0x1030` is the I/O port of the PCI serial port. You can get this value by typing:

```
# lspci -v | grep -A 5 "Communication controller"
```

After rebooting, you can connect to the console at the other end of the serial cable using "`screen /dev/ttyS0 115200`"

6. Reboot the machine, and choose the new entry in GRUB's menu.

```
# reboot
```

Don't forget to select "`XEN 3.3 with Fedora 8 (2.6.21.7-5.fc8xen)`" after rebooting

7. Login as root and check whether Xen and Domain 0 are running

```
# uname -r
```

Make sure the result is "`2.6.21.7-5.fc8xen`"

```
# xend start
# xm list
```

If you can see the following lines, it means that Xen and Domain 0 have been successfully installed.

```
    Name                                      ID   Mem VCPUs      State   Time(s)
    Domain-0                                   0  1024     1     r-----     642.6
```

3.Installing para-virtualized guest domains

3.1. Preparing guest's file system

1. Login to a shell as `root`

2. Prepare a guest root partition.
   You may create an empty partition using `cfdisk` or `lvcreate`. The size of the guest partition should be large enough to accommodate all existing files in the Domain 0 (At least 10 GB). To create an LVM volume for the guest, use the following commands: (Think carefully before you type `/dev/sda5`. If unsure, please go back and check step 1.1-5.)
   ```
   # vgcreate VG0 /dev/sda5
   # lvcreate -L 10G -n dom1 VG0
   ```
   Make sure the volume group and a new volume is created:
   ```
   # ls -al /dev/VG0/dom1
   ```
   In the remaining of this section, we assume the device filename of the guest root partition is `/dev/VG0/dom1`

3. Format and mount the guest partition (DO remember to replace `/dev/VG0/dom1` with your actual device file!)
   ```
   # mkfs -t ext3 /dev/VG0/dom1
   # mount /dev/VG0/dom1 /mnt
   ```

4. Clone the guest file system from Domain 0
   ```
   # cp -ax /{home,bin,dev,etc,lib,lib64,root,sbin,usr,var} /mnt
   # mkdir /mnt/{media,misc,mnt,net,opt,proc,selinux,srv,sys,tmp}
   ```
   This step can take a long time. Be patient!

5. Modify `fstab` for the guest domain (Never forget typing `/mnt`)
   ```
   # vi /mnt/etc/fstab
   ```
   Remove any unused lines in the file, especially the ones that describe the mount points for `/`, `/boot` and `swap`. Then append the following line:
   ```
       /dev/hda1            /                   ext3    defaults        1 1
   ```
   The above `fstab` line tells the guest OS to mount `/dev/hda1` as its root directory. `/dev/hda1` is the virtual device name as seen by the guest domain, which is actually `/dev/VG0/dom1` on the physical machine. The map is established in the domain's configuration file, which we will create later.

6. Unmount the guest root partition
   ```
   # umount /mnt
   ```

Note:
For more information about how to use a physic partition, an LVM volume, a loop device or NFS as the guest's storage, please refer to Xen 3.0 User's Manual. (http://bits.xensource.com/Xen/docs/user.pdf)

3.2. Booting the guest domain

1. Create a new init-ram-disk (initrd) for guest domains by adding the front-end drivers
   ```
   # mkinitrd -v -f --with=ext3 --with=xenblk --with=xennet --with=netloop /boot/
   initrd-xenU  2.6.21.7-5.fc8xen
   ```
   Check that the file "`/boot/initrd-xenU`" exists before moving on.

2. Create the domain's configuration file
   ```
   # vi dom1.conf
   ```
   An example for its content is:
   ```
       name='dom1'
       kernel = '/boot/vmlinuz-2.6.21.7-5.fc8xen'
   ```

```
    ramdisk = '/boot/initrd-xenU'
    memory=1024
    vif=['']
    disk=["phy:/dev/VG0/dom1,hda1,w"]
    root='/dev/hda1 ro'
```

In this file, we tell Xen what the domain's name is, which kernel image and init-ram-disk it uses, how much memory it has, and also what disk image it uses. The last two lines are the most important. They show that the physical device `/dev/VG0/dom1` is assigned to `dom1` as its `/dev/hda1`, with writing permissions. The last line "`root='/dev/hda1 ro'`" is the parameter to be passed to the guest's kernel. The two lines must match the guest domain's `/etc/fstab` file as described in step 3.1-5.

3. ***(Optional)*** Assign a swap partition for the guest domain

```
# lvcreate -L 2G -n dom1swap VG0
# mkswap /dev/VG0/dom1swap
# vi dom1.conf
```

Change the line starts with "`disk`" to:

```
    disk=["phy:/dev/VG0/dom1,hda1,w", "phy:/dev/VG0/dom1swap,hda2,w"]
```

Modify `fstab` for the guest domain

```
# mount /dev/VG0/dom1 /mnt
# vi /mnt/etc/fstab
```

Append a line:

```
    /dev/hda2          swap                    swap    defaults        0  0
```

```
# umount /mnt
```

In this way, the volume `/dev/VG0/dom1swap` is assigned to `dom1` as its virtual device `/dev/hda2`, and the virtual device will be mounted as a swap partition at guest startup.

4. Boot the guest domain

```
# xm create dom1.conf
```

You should see:

```
    Using config file "./dom1".
    Started domain dom1
```

```
# xm list
```

`dom1` should appear in the domain list.

5. To connect to a guest domain's console, type:

```
# xm console dom1
```

To leave the guest's console, press `Ctrl+]`

You can also use "`xm create -c`" to automatically connect to the domain's console at the time of creation.

6. To shutdown the guest domain, type:

```
# xm shutdown dom1
```

Note:

For more information about the domain configuration file, please refer to `/etc/xen/xmexample*`;

Note:

If the Domain 0 or the guest domain failed to boot, the following log files may help you to diagnose the problem:

```
    /var/log/boot.msg
    /var/log/boot.omsg
```

```
/var/log/messages
/var/log/xen/xend.log
/var/log/xen/xen-hotplug.log
/var/log/xen/xend-debug.log
```

4.Installing HVM domains

4.1. Pre-requirements

1. Make sure you have had these packages installed: `dev86`, `LibVNCServer`, `SDL-devel`. If not sure, try the following commands:
   ```
   # yum install dev86
   # yum install libvncserver vnc
   # yum install sdl*
   ```
2. Make sure the file `/usr/lib/xen/boot/hvmloader` exists. If not, try to make it again.
   ```
   # cd xen-3.3.0
   # make -C tools/firmware/hvmloader install
   ```
3. Make sure the hardware and the BIOS have had HVM support enabled.
   ```
   # xend start
   # xm info | grep xen_caps
   ```
   You should see the output contains `hvm-3.0-*`. Otherwise, check the CPU feature and BIOS settings.
4. Get an OS install image.
   Here we assume we are going to install Windows Server 2003 and the installation CD image file is `/opt/win2k3.iso`

4.2. Installing an HVM guest

1. Prepare a guest disk image.
   You may create an empty partition using `cfdisk` or `lvcreate`, or using a sparse file as the guest's disk image. This step demonstrates how to create a sparse file.
   ```
   # dd if=/dev/zero of=/opt/hvm.img bs=1M count=1 seek=10239
   ```
   Here, "`seek=`" specifies the size of the disk image, in megabytes.
   In the remaining of this section, we assume the filename of the guest disk image is `/opt/hvm.img`

2. Create domain configuration file. You may start with Xen's sample:
   ```
   # cp /etc/xen/xmexample.hvm hvm1.conf
   # vi hvm1.conf
   ```
   Modify the following lines as necessary:
   ```
       memory = 512
       name = "hvm1"
       disk = [ 'file:/opt/hvm.img,hda,w', 'file:/opt/win2k3.iso,hdc:cdrom,r' ]
       boot = "d"
       monitor = 1
   ```
   Do not forget to replace `/opt/hvm.img` and `/opt/win2k3.iso` with the actual image filename.
   Note that "`boot = "d"`" tells Xen to boot from CD-ROM first. It is required for installation from CD image. After installation, you should change it to "`boot = "c"`"

3. Startup the HVM domain and begin the OS installation
   ```
   # xend start
   # xm create hvm1.conf
   ```
   To view the installation GUI, you should run the following command in X window:

```
# xm vncviewer hvm1 &
```

You can also starts the VNC viewer at the time of domain creation by:

```
# xm create hvm1.conf --vncviewer
```

Hint: you can also specify "`sdl = 1`" in domain's configuration file to avoid attaching to VNC every time.

4. Follow the screen instruction to finish the guest installation. If the virtual machine reboots, you can re-attatch to its screen using "`xm vncviewer`" command.

5. After installation, change the boot order in `hvm1.conf` to "`boot = "c"`", and then reboot the guest domain.

5.Compiling para-virtualized kernels from the source

Sometimes it is desirable to compile the Domain 0 kernel from the source code, for example, when you would like to change the kernel configuration, or modify some para-virtualized drivers.

5.1. Getting the source code

1. Install kernel builder tools

```
$ su -c "yum install yum-utils rpmdevtools"
```

2. Get the source rpm

```
$ cd
$ rpmdev-setuptree
```

You should have a directory "`rpmbuild`" appeared in the current directory. It is required for extracting the kernel source

```
$ yumdownloader --source kernel-xen
```

This command downloads the newest source rpm of the Domain 0 kernel into the current directory. The file name of the rpm should be "`kernel-xen-<version>.src.rpm`"

```
$ su -c "yum-builddep kernel-xen*.src.rpm"
$ rpm -Uvh kernel-xen*.src.rpm
```

The previous command writes the rpm content into the build directory (`rpmbuild`). You can ignore any warnings.

3. Extract the kernel source

```
$ cd rpmbuild/SPECS
$ rpmbuild -bp --target=`uname -m` kernel.spec
```

The kernel source tree is now located in the `./rpmbuild/BUILD/kernel-<version>/linux-<version>.<arch>` directory.

5.2. Configuring and making the kernel

1. Configure the kernel

```
$ cd rpmbuild/BUILD/kernel-*/linux-*
$ make menuconfig
```

You can also manually edit the configuration file by

```
$ vi .config
```

For example, if you would like to include `extfs` and para-virtualized drivers into the kernel (rather than compile them as modules), you can change the following lines to "`y`"

```
    COFNIG_EXT2_FS
    CONFIG_EXT3_FS
    CONFIG_XEN_*
```

2. Make and install the kernel

   ```
   $ make
   $ sudo make modules_install install
   ```

3. Create the init-ram-disk for the kernel, assuming the compiled kernel version is "`2.6.21.7`"

   ```
   # mkinitrd -v -f /boot/initrd-2.6.21.7.img 2.6.21.7
   ```

   Here, we assume you have built the para-virtualized drivers into the kernel in step 5.2-1. Otherwise, you have to use the following command:

   ```
   # mkinitrd -v -f --with=ext3 --with=pciback --with=netbk --with=blkbk --
   with=netloop --with=xenbus_be --with=netloop --with=xenblk --with=xennet /
   boot/initrd-2.6.21.7 2.6.21.7
   ```

4. Create a new boot entry in GRUB. Please refer to step 2.2-4. Don't forget to replace the filename of the kernel and init-ram-disk.

5. Reboot to the new kernel

6. ***(Optional)*** To use the new kernel in guest domains, you need to copy the modules to the guest root partition.

   ```
   # mount /dev/VG0/dom1 /mnt
   # cp -ax /lib/modules/2.6.21.7 /mnt/lib/modules
   # umount /mnt
   ```

Note:

For more information, please refer to http://fedoraproject.org/wiki/Docs/CustomKernel